Off-line performance maximisation in feed-forward neural networks by applying virtual neurons and covariance transformations

Cesare Alippi - Raffaele Petracca - Vincenzo Piuri
Dipartimento di Elettronica -Politecnico di Milano
Piazza L. Da Vinci 32, 20133 Milano, Italy

ABSTRACT

Optimisation of a feed-forward neural paradigm for a given application involves problems such as maximisation of the generalisation ability (relevant to provide effectiveness) and structure minimisation (allowing for physical realisability by using dedicated VLSI devices). This paper proposes a contemporaneous solution of these conflicting goals. The globally-optimised structure is identified by using a covariance matrix transformation and layers of virtual neurons.

I. INTRODUCTION

Traditional techniques for configuring a feed-forward network model on a specific application are directed to identify the "best" values of the interconnection weights for a given network structure. The number of layers and the cardinality of each layer are predefined by the designer according to his expertise without any general theoretical rule. Optimisation is concerned with the evaluation of the weights' configuration that minimises the training error. When the neural paradigm has been configured, it can be mapped onto the hardware architecture.

Suitable techniques must then be applied to such an architecture in order to reduce the layers' complexity, in terms of number of neurons/connections, and make feasible the dedicated VLSI implementation. Generally, the training procedure needs to be repeated whenever neurons have been removed; if only interconnections are pruned, topologies with different neurons' fan-in may result which prevents a compact and effective implementation. Appropriate techniques may be used to improve the generalisation ability of the configured structure as a separate optimisation step [1, 2].

Research is also concerned with other behavioural aspects of the neural paradigms, e.g., with generalisation ability and sensitivity. Generalisation is mainly directed to reduce the recall error and maximise performances. The sensitivity issue is related to minimisation of the impact of errors at network's outputs in presence of computational errors within neural operators; this is fundamental to provide the network with a fault-tolerance degree.

Global optimisation of the feed-forward paradigm, which includes generalisation maximisation and structure minimisation, is a complex multiple-goal problem. This paper present a novel approach to the contemporaneous solution of these conflicting goals. Basically, our technique maximises the generalisation ability and, by exploiting the insensitivity properties, allows identification of the neurons to be removed.

The neural paradigm must be configured for the specific application as in the traditional approaches. However, the designer is allowed to exceed largely with the number of neurons without taking particular care of the mandatory minimisation of the learning error (i.e., of the length and the complexity of the training procedure), guaranteeing the convergence of the minimisation algorithm and the computation ability.

Our methodology evaluates the covariance matrix associated with each hidden layer of the neural network and creates one additional virtual layer (composed by virtual neurons) for each of them just for optimisation purposes. The task is accomplished by applying a transformation of each interconnection weights based on the analysis of the eigenvalues spectrum of the related covariance matrix. This leads to redistribute the neural computation in order to maximise the generalisation ability and the insensitivity. The final network structure is then derived by fusing each virtual layer with the corresponding nominal ones without requiring any further training. If the network structure can be mapped onto the given hardware architecture or implemented in software as efficiently as required by the application, no further design activity is necessary. As a side effect, maximisation of the insensitivity degree implies maximisation of the intrinsic error masking capability and, in particular, of the ability of tolerate errors due to faults in the hardware architecture or to finite precision operations.

Conversely, if the realisation constraints are not satisfied, our methodology provides basic information on how to reduce the neural structure while preserving performances. In particular, it allows to exactly identify the neurons with a small influence on the neural operations; these neurons can be removed with marginal loss in performance.

Therefore, the proposed methodology is directed to optimise the generalisation ability, but - at the same time - provides information for optimal minimisation of the network structure and increases its insensitivity degree.

0-7803-2570-2/95 $4.00

II.VIRTUAL LAYERS AND THE COVARIANCE MATRIX TRANSFORMATION

Let us consider a generic layer of a successfully trained feed-forward neural networks. For instance, the hidden layer of a three-layered neural network of regression type (a network composed of a single hidden layer with a non linear sigmoidal-like activation function and a linear output neuron); generalisation to multi-output topologies is straightforward.

Let us denote with x_{pi} the output of the i-th hidden neuron generated by p-th training sample. The *covariance matrix* C associated with the layer and evaluated over the training set can be expressed as:

$$C = \left[c_{ij} = \frac{1}{N} \sum_{p=1}^{N} \left(x_{pi} - m_i \right) \left(x_{pj} - m_j \right) \right] \qquad (1)$$

being $m_i = 1/N \sum_{p=1}^{N} x_{pi}$ the mean output value of the generic neuron, and N the cardinality of the training set. The covariance matrix is obviously symmetric, semi-definite positive and, therefore, can be diagonalised by means of the orthonormal matrix U :

$$C = U \, diag(\lambda_1, \ldots, \lambda_n) \, U^T \qquad (2)$$

Topological optimisation at the neuron level can now be achieved by analysing the eigenvalues distribution as suggested in [1]; in more detail, there it is assumed that the number of neurons to be removed equalises the number of less relevant eigenvalues (the training procedure iterates until performances improve). It should be noted that after eliminating some hidden units according to the procedure suggested in [1] the network needs to be retrained since the method does not provide any correspondence between the neurons to be removed and the less relevant eigenvalues.

In this paper we suggest a different optimising procedure which can be used to

1. increase performaces without the need of retraining. This is done by analysing the eigenvalues spectrum of the covariance matrix and applying a transformation to the network's weights which re-distributes their information content. Information redundancy is then eliminated; the effective number of degrees of freedom decreases maintaining the same topological complexity.
2. reduce the complexity of the network topology. After having applied step 1., spatial redundancy, in terms of the number of neurons/connections, is eliminated by means of pruning tecniques; at the end, the topological complexity of the network matches the effective information content.

The goals can be achieved by introducing the concept of virtual layers. A virtual layer is a layer between two consecutive real layers whose neurons have identity activation functions and weights connecting the real neurons such that the covariance matrix associated with the virtual layers' outputs is diagonal.
For instance, and referring to figure 1, the matrices defining the interconnections between the virtual layer and the real ones can be chosen as U and $U^T W$, being W the weights matrix associated with the real layers. Since $W = U\left(U^T W\right)$, the virtual layer has not modified the network's behaviour.

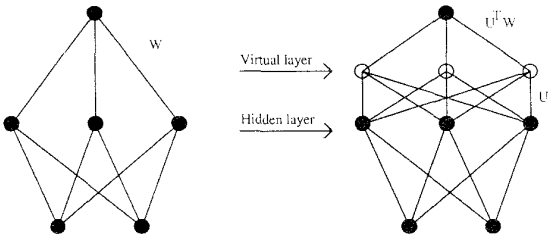

Figure 1: Introducing the virtual layer

It can be proved from (2) that the covariance matrix C_V associated with the outputs of the virtual layer is diagonal: $C_v = diag\left(\lambda_1, \ldots, \lambda_n\right)$. In other words, each virtual neuron has variance equal to the corresponding eigenvalue while the covariance among different neurons is null. Removal of a generic eigenvalue is obtained by forcing it to zero. The output of the corresponding virtual neuron becomes its mean value; this constant can be taken into account by modifying the biases of subsequent neurons, thus allowing removal of the virtual neuron. The new weights matrix W' can now be easily determined as $W' = U'\left(U'^T W\right)$ being U' obtained from U by setting to zero the rows associated with the removed eigenvalues.

III. IMPROVING PERFORMANCE BY REMOVING VIRTUAL NEURONS

The substitution of a not relevant virtual neuron with its mean value determines a small and insignificant increase in the training error. On the other hand, this creates the relationship $W = U'\left(U'^T W\right)$ among the network's weights. This relationship reduces the network's effective number of degrees of freedom and consequently the Vapnik-Chervonenkis dimension [2].

If the information being removed is attributable to noise, the network performance is improved. In fact, reduction of not relevant eigenvalues intuitively coincides with reduction of the impact of noise on performance (overfitting effects). This has always been verified in our applications of non-linear regression.
Let us consider, as an example, the problem of approximating the non linear function

$$y = -x\sin(x^2) + \frac{e^{-0.23x}}{1+x^4} \qquad (3)$$

with a neural network of regression type. A training set of *N=40* data was extracted from the [-2, 2] interval, while each datum was corrupted with an additive gaussian noise GN(0, 0.04).
The goal is to determine an optimal network in the generalisation ability sense, able to infer, from noisy data, the real function given in expression (3).
In figure 2 the real function is plotted with a continuous line, the ten hidden units best neural approximator with a dashed line, and the training data with circles. The network is overdimensioned and in evident overfitting in the -1 neighbourhood.

Performance improvement can now be achieved by computing the covariance matrix according to equation (1) and introducing the virtual layer as shown in figure 1. The eigenvalues distribution of the covariance matrix is plot in figure 3a with a semi-logarithmic scale. In the counterpart figure 3b, it is shown the validation error (we used the classic mean squared error both for training and validation) computed over 134 new data as a function of the number of the removed virtual neurons. It can be seen that the best performances are obtainable by eliminating two virtual neurons. The network was then modified by removing the two less-relevant virtual neurons; the correspondent final performances are plotted in figure 2b. As expected, overfitting effects are now reduced and the network has improved its performance; in this application we verified that the generalisation error was very closed to its minimum. Anyway, the network is still redundant since the number of degrees of freedom is greater than the one of the optimal topology having just five neurons. We have to

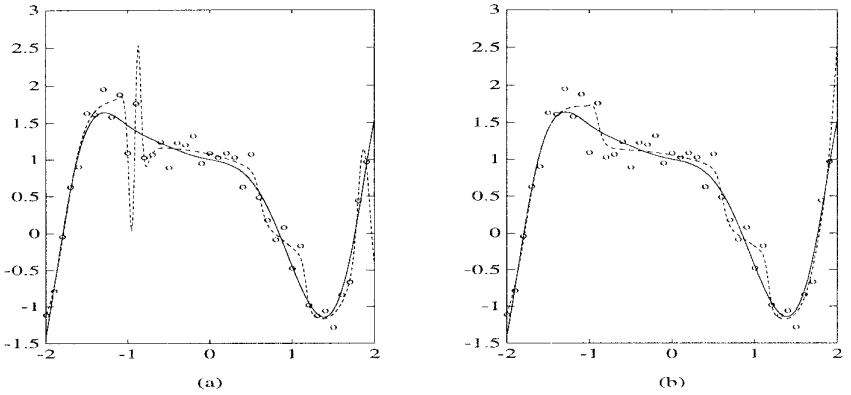

Figure 2: Performance before removing virtual neurons (a) and after removal of 2 virtual neurons (b)

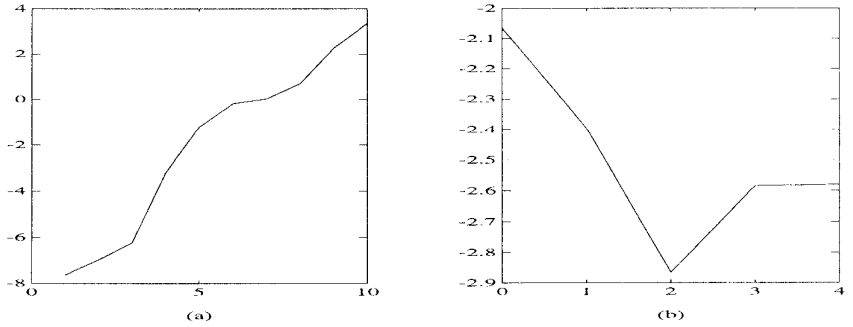

Figure 3: A semilog plot of the C's eigenvalues (a) and the validation error as a function of the number of the removed virtual neurons (b)

outline that at this level the network has improved its generalisation performance but it is redundant. Topological minimisation procedure has to remove the unexploited degrees of freedom and it will be presented in the subsequent section.

IV. THE OPTIMISING PROCEDURE

Topological redundancy can be shown with a sensitivity analysis. In our examples, we consider the increase in the training error correspondent to two experiments:
1) removal of the generic *i*-th hidden neuron;
2) removal of the generic *i*-th hidden neuron and consequent rearrangement of the weights configuration according to an extension of the Optimal Brain Damage -OBS- technique [3].

It can be seen (figure 4a) that removal of a hidden unit followed by the OBS technique provides an increase in the training error (* points) on the average smaller than the correspondent one obtained by removing the same neuron without any weights rearrangement (x points). In both cases the insensitivity improves whenever the experiments are carried out after removal of the appropriate number of virtual neurons.This can be seen in figure 4b.

The covariance trasformation and virtual neurons removal have a positive effect on sensitivity issues. In particular, the network becomes less sensitive to removal of hidden units, thus proving that those degrees of freedom are not relevant to final performance. The network is therefore topologically redundant.
Finally, we can identify the optimal network in the generalisation ability sense by iterating the procedure based on covariance transformation, virtual layers and pruning techniques.

The algorithm can be summerised as follows:

1. Training phase;
2. Covariance matrix evaluation and elimination of virtual neurons until performances improve;
3. Application of a pruning technique, an OBS like procedure for instance;
4. Training phase;
5. The procedure iterates from step 2 until performances improve.

At the end of the procedure we obtain the optimal network topology which matches performance maximisation and minimal complexity.

If the optimising procedure considers just elimination of virtual neurons, the provide neural structure has both incresed performances and its insensitivity degree, in the sense that it is able to minimise the errors at the networks' outputs caused by errors affecting the neural computation. If the whole procedure is considered, as suggested by the algorithm, the final network is optimal in the generalisation ability sense and compact thus making feasible its VLSI implementation.

V. REFERENCES

[1] A.S.Weigend, D.E.Rumelhart. (1991) *The effective dimension of the space of hidden units*. In Proc. of IJCNN Singapore.

[2] I.Guyon, V.Vapnik, B.Boser, L.Bottou, S.A.Solla. (1992) *Structural risk minimization for character recognition*. In NIPS4, CA: Morgan Kaufmann.

[3] C.Alippi, R.Petracca., V.Piuri (1995) *Covariance transformation to maximise the generalisation ability in regression type neural networks*. Internal report Dip. Elettronica, Politecnico di Milano, Milano, Italy.

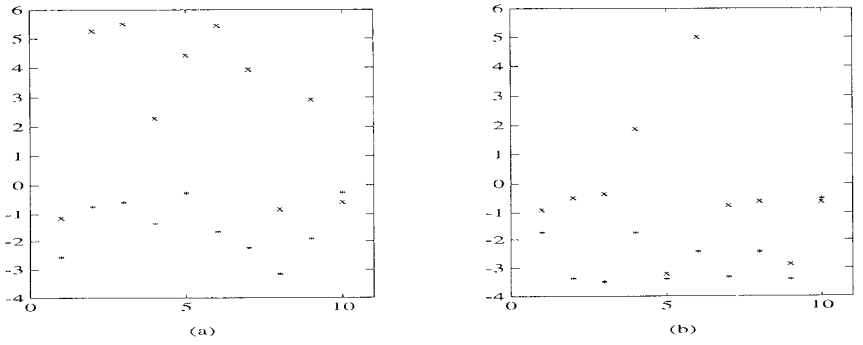

Figure 4: Semilog plot of the increase of training error consequent to the elimination of a hidden units for experiment 1 (a) and experiment 2 (b)